Distributed decision fusion under unknown distributions

Nageswara S. V. Rao
Oak Ridge National Laboratory
Center for Engineering Systems Advanced Research
Oak Ridge, Tennessee 37831-6364
E-mail: raons@ornl.gov

S. Sitharama Iyengar
Louisiana State University
Department of Computer Science
Baton Rouge, Louisiana 70803

Abstract. The problem of distributed decision fusion is studied in the case when the probability distributions of the individual detectors are not available. The detector system is available so that a training sample can be generated by sensing objects with known parameters or classification. Earlier solutions to this problem required some knowledge of the error distributions of the detectors, for example, either in a parametric form or in a closed analytical form. Here we present three methods that, given a sufficiently large training sample, yield an approximation to the optimal fusion rule with an arbitrary level of confidence. These methods are based on (i) empirical estimation, (ii) approximate decision rule, and (iii) nearest-neighbor rule. We show that a nearest-neighbor rule provides a computationally viable solution, which approximates a neural network-based one while ensuring fast computation. © *1996 Society of Photo-Optical Instrumentation Engineers.*

Subject terms: sensor fusion; distributed decision fusion; empirical estimation; nearest-neighbor rule; neural networks; Lipschitz functions.

Paper SF-008 received June 29, 1995; revised manuscript received Sep. 1, 1995; accepted for publication Oct. 3, 1995.

1 Introduction

The problem of fusing decisions made by individual agents has been extensively studied in areas such as political economy,[1] reliability,[2] forecasting,[3] pattern recognition,[4] neural networks,[5] and decision fusion.[6] In the well-studied area of decision fusion, the basic problem is to combine the decisions made by a number of distributed detectors.[7–10] A typical fusion rule in this case is in the form of a Bayesian rule[7] or Neyman–Pearson test,[9,10] and requires the knowledge of underlying error probability densities. Furthermore, analytical expressions for the error densities must be in a convenient form to ensure reasonable computational speeds. In many detection systems, it might be possible to utilize the knowledge about the system to obtain the required probability densities. In turn, this knowledge could be based on the experience with the system, possibly in the form of empirical data observed during experimentation or operation. Thus such an approach involves inferring densities from empirical data. In the distributed decision fusion context (and in several other contexts[11]), the density estimation is harder than the problem of estimating a fusion rule (or function in general) directly from empirical data: the latter involves estimation over a smaller class of functions, e.g., a set of Boolean functions in the present problem, whereas the former involves estimation over a much larger class of densities.

In this paper, we study the case when no information about the probability distributions is available (along the lines of Ref. 12). Our solution relies on utilizing a training sample. We obtain the sample sizes required to arbitrarily bound the probability of disagreement between a fusion rule computable when the error densities are known and its empirical implementation based on a sample.

The proposed technique is to be used mainly when accurate estimates of the probabilities are either not available or are computationally difficult. For a system of independent detectors, if the exact analytical form of the probabilities is available, the methods of Refs. 1 and 9 could be used to implement the required fusion rule. Also, the sample-based approach could be useful from an additional viewpoint. In some cases—typically in a system of nonindependent detectors—even if the distributions are available, it could be computationally intractable (NP-complete) to implement a Bayesian test.[13,14] In such cases, Monte Carlo simulation can be used to generate the empirical data and the methods proposed here can be used to implement a possibly suboptimal fusion rule (as illustrated in the example of Sec. 3).

Consider that the data vector $x \in \Re^d$ is produced according to the distribution P_x. Let a probabilistic rule yield hypothesis $h \in \{H_0, H_1\}$ based on data vector $x \in \Re^d$ according to a probability distribution $P_{h|x}$. For example, in object detection systems, x corresponds to readings taken by a sensor system and h is the decision where the hypothesis H_0 (or H_1) corresponds to the presence (or absence) of the required object. Consider a parallel suite[6] of N detectors (see Fig. 1) such that corresponding to data vector $x \in \Re^d$, the i'th detector outputs $y^{(i)} \in \{H_0, H_1\}$ according to an unknown probability distribution $P_{y^{(i)}|x}$. Each detector D_i, for $i = 1,2,\ldots,N$, makes a decision $y^{(i)} \in \{H_0, H_1\}$, and the fu-

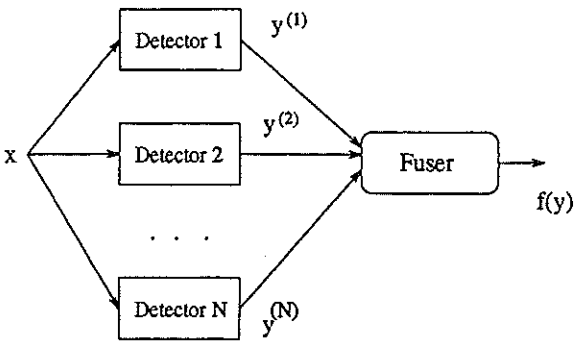


Fig. 1 Parallel sensor suite.

sion center receives the decision vector $y=[y^{(1)},y^{(2)},\ldots,y^{(N)}]$ and outputs either H_0 or H_1. The fuser design calls for a method (typically a function) that combines the outputs of various detectors and returns a "consolidated" hypothesis while minimizing a suitable cost function such as expected error. Thus our objective* is to choose a fusion rule $f:\{H_0,H_1\}^N \mapsto \{H_0,H_1\}$ from a family of functions $\mathscr{F}$ to minimize the expected error given by

$$I(f)=\sum_{y\in\{H_0,H_1\}^N}\ \sum_{h\in\{H_0,H_1\}}\ \int_x [f(y)\oplus h]\,dP_{y|x}\,dP_{h|x}\,dP_x, \tag{1}$$

where $y=(y^{(1)},y^{(2)},\ldots,y^{(N)})$ is conditionally distributed according to an unknown $P_{y|x}$, and

$$a\oplus b=\begin{cases}0 & \text{if } a=b\\ 1 & \text{otherwise.}\end{cases}$$

In general, $\mathscr{F}$ can consist of all possible functions of the form $f:\{H_0,H_1\}^N \mapsto \{H_0,H_1\}$ in which case $|\mathscr{F}| = 2^{2^N}$. For example in neural network methods, $\mathscr{F}$ may correspond to a set of all neural networks of a fixed architecture (fixed number of hidden layers and fixed connections). The minimization problem in such case is to choose weights that minimize Eq. (1) for a network of the chosen architecture.

In object recognition systems, each detector could base its decision on possibly different object features, i.e., each detector D_i may be sensitive to only certain components of x. A number of formulations along the lines of Eq. (1) have been studied (see Ref. 6 for a comprehensive treatment) under various conditions. One of the earliest formulations deals with independent detectors in which case the fusion rule takes the form of an easily computable test (e.g. Bayesian rule or Neyman–Pearson test expressed in terms of the densities of various detectors[7]). Most existing solutions to this class of problems require knowledge about the error densities, with the exception of some recent results.[12] Here we consider the case of unknown distributions, where the information from the system is in the form of a training sample.

A training l-sample $(x_1,h_1,y_1),(x_2,h_2,y_2),\ldots,(x_l,h_l,y_l)$ is given where $y_i=[y_i^{(1)}, y_i^{(2)},\ldots,y_i^{(N)}]$ and $y_j^{(i)}$ is the output of D_i in response to input $x_j\in\Re^d$. Informally, the training example (x_i,h_i,y_i) gives the data vector x_i, the correct hypothesis h_i, and the vector y_i of hypotheses of the detectors. The problem is to estimate a fusion rule $f:\{H_0,H_1\}^N \mapsto \{H_0,H_1\}$, based on the sample, such that $f(y)$ "closely" approximates h corresponding to x.

Let $f^*\in\mathscr{F}$ minimize $I(f)$ in Eq. (1) (it is assumed that $\mathscr{F}$ is suitably chosen such that f^* exists[15]). In our formulation, f^* cannot be computed since the underlying distributions are unknown. Furthermore, since no restrictions are placed on the distributions, it will not be possible to infer the function f^* (with probability one) based on only a finite sample. Typically, it is only possible to compute an approximation to f^* with some probabilistic guarantees.

We consider three types of methods to compute an approximation to f^*: (i) empirical estimation, (ii) approximate decision rules, and (iii) the nearest-neighbor rule. These methods are based on well-known techniques but their applicability and performance is to be examined in detail for the case of a distributed decision problem. The first and second methods are general but do not yield conveniently computable approximations and sample sizes respectively. The third method provides a computationally conducive method but is not as general (in that $\mathscr{F}$ is restricted to a class of Lipschitz functions). Further, the last method also provides an approximation to a neural network solution without incurring the high computational cost for the training of the latter. Special attention is paid to (feedforward) neural network methods in this paper, since the fusion rule estimation problem is unlikely to be handled effectively by linear methods and neural networks have been found to be effective in a number of nonlinear function estimation problems. But the training problem of neural networks is particularly difficult in a number of situations. For the present problem, our results indicate that the nearest-neighbor rule provides a good alternative to neural networks.

The present formulation has been motivated by the sensor fusion problems that arise in robotic applications,[16] where the individual sensors have been built and mounted on the robot. Obtaining accurate probabilistic models of the sensors is a more challenging task than performing experiments by sensing a set of objects with known features (the situation in other applications, however, could be significantly different). Here, the training samples can be obtained by sensing the objects that belong to given classes and also the objects that do not. Distributed decision fusion problems with unknown probabilities have been studied in Refs. 12 and 16; both methods are based on estimating the probabilities from a sample, but the latter provides sample size estimates that guarantee specified levels of confidence. The general problems of estimating functions based on finite sample have been extensively studied.[11,15] Recently, the computational aspects of such problems have been receiving increasing attention.[17,18]

*Only the fusion problem is addressed here in that the local detectors are given, i.e., we do not address the problem of designing the local detectors.

Three solutions to the decision fusion problem are presented in Sec. 2. An example is discussed in Sec. 3.

2 Fusion Rule Estimation

We first consider the case when $\mathscr{F}$ is the set of Boolean functions using the methods of empirical estimation and approximate decision rules. Then we restrict $\mathscr{F}$ to a certain class of Lipschitz functions and show that nearest-neighbor rules provide a computationally viable option.

2.1 *Empirical Risk Minimization*

We apply the empirical risk minimization method of Vapnik[8] specialized to the present problem. The $I(f)$ of Eq. (1) can be rewritten as

$$I(f)=\int_{y,h,x}[f(y)\oplus h]dP_{y,h,x},$$

where $dP_{y,h,x}$ corresponds to $dP_{h|x}dP_{y|x}dP_X$. Now consider that the empirical estimate

$$I_{\text{emp}}(f)=\frac{1}{l}\sum_{i=1}^{l'}[f(y_i)\oplus h_i]$$

is minimized by $f_{\text{emp}}\in\mathscr{F}$. Recall that f^* that minimizes $I(.)$ cannot be computed since $P_{y,h,x}$ is not known, whereas f_{emp} can be computed since $I_{\text{emp}}(.)$ depends only on the sample. If $\mathscr{F}$ is the set of all feedforward neural networks of a fixed architecture, the computation of f_{emp} is called the training problem. We show in the following theorem that for a sufficiently large sample

$$P[I(f_{\text{emp}})-I(f^*)>\epsilon]<\delta$$

for arbitrarily specified $\epsilon>0$ and δ, $0<\delta<1$. Thus this approach yields a fusion rule f_{emp} whose "error" is bounded (within minimum possible error) by an arbitrarily specified precision ϵ with arbitrarily specified confidence $1-\delta$ given a sufficiently large sample.

Theorem 2.1 Let $\mathscr{F}$ denote the set of all Boolean functions of N variables, and let $f_{\text{emp}}\in\mathscr{F}$ minimize $1/l\Sigma_{j=1}^{l}[f(y_j)\oplus x_j]$. Then we have $P[I(f_{\text{emp}})-I(f^*)>\epsilon]<\delta$ given a sample of size

$$l=\frac{2}{\epsilon^2}[2^N\ln 2+\ln(2/\delta)].$$

Also, $I(f_{\text{emp}})\to I(f^*)$ with probability one as $l\to\infty$.

Proof. The first part is a direct consequence of Theorem 6.1 of Vapnik[11] (see also Ref. 16) which shows that

$$P[I(f_{\text{emp}})-I(f^*)>\epsilon]<2^{1+2^N}e^{-\epsilon^2 l/2}$$

by noting that $\mathscr{F}$ contains 2^{2^N} elements. For the second part, consider for any $\epsilon>0$

$$\begin{aligned}\sum_{l=1}^{\infty}P[I(f_{\text{emp}})-I(f^*)>\epsilon]&\leq 2^{1+2^N}\sum_{l=1}^{\infty}e^{-(\epsilon^2 l/2)}\\&\leq 2^{1+2^N}\int_0^{\infty}e^{-(\epsilon^2 x/2)}dx\\&\leq\frac{2^{2+2N}}{\epsilon^2},\end{aligned}$$

which is finite. Thus the result follows from the Borel–Cantelli lemma.[19] □

This result can be perceived as an existence proof for the overall tractability of the present formulation of the distributed decision problem. However, a practical implementation of the solution is computationally difficult. In general, the problem of computing f_{emp} is NP-hard and could require a time complexity of $O(2^{2^N})$ for the sample size estimated above (with fixed ϵ and δ). A generalization of the above method for the case of h taking continuous values and $\mathscr{F}$ having nonfinite cardinality can be found in Ref. 16.

2.2 *Approximate Decision Rule*

We consider a technique that utilizes the solutions to the distributed decision fusion problems applicable when the error densities are known. Typically in these cases, the solution is in the form of a probability ratio test[20] expressed in terms of the various densities, which are unknown in the present formulation. Here we consider an approximation of the test based on a sample.

We first show that the regression function $E(h|y)$ minimizes Eq. (1)[11] and then show that an approximation to it can be computed using the sample (see Refs. 21 and 22 for the solutions of the latter type). Consider the following:

$$\begin{aligned}I(f)&=\int_{y,h,x}[f(y)\oplus h]dP_{y,h,x}\\&=\int_{y,h,x}[f(y)-h]^2dP_{y,h,x}\\&=\int[f(y)-E[h|y]]^2dP_{y,h,x}+\int(E[h|y]-h)^2\\&\quad\times dP_{y,h,x}+2\int[f(y)-E[h|y]]\\&\quad\times(h-E[h|y])dP_{y,h,x}\\&=\int[f(y)-E[h|y]]^2dP_{y,h,x}\\&\quad+\int(E[h|y]-h)^2dP_{y,h,x}.\end{aligned}$$

Note that the second term is independent of f and hence the minimum is achieved by the regression function. It is well known that f^*_{adr} given by

$$f^*_{\text{adr}}(y)=\begin{cases}H_0 & \text{if } P_{y,h,x}(H_0|y)>P_{y,h,x}(H_1|y)\\ H_1 & \text{otherwise}\end{cases}$$

minimizes the first term. This function in turn can be expressed as a test checking the condition $P_{y,h,x}(H_0 \cap y) > P_{y,h,x}(H_1 \cap y)$. Now consider an empirical implementation f_{adr} of f^*_{adr} defined as

$$f_{\text{adr}}(y) = \begin{cases} H_0 & \text{if } \hat{P}_{y,h,x}(H_0 \cap y) > \hat{P}_{y,h,x}(H_1 \cap y) \\ H_1 & \text{otherwise} \end{cases}$$

where $\hat{P}_{y,h,x}(H_i \cap y)$ be the fraction of sample points of the form (y, H_i) in the sample for $i=0,1$. By Theorem 3.1 of Ref. 22 (which can also be derived from Lemma 3.1 of Ref. 21), we have $P[f^*_{\text{adr}}(y) \neq f_{\text{adr}}(y)] < \delta$ for the sample size l given in the following theorem.

Theorem 2.2 Let $\mathscr{F}$ denote the set of all Boolean functions of N variables, and let $\hat{P}_{y,h,x}(H_i \cap y)$ be the fraction of sample points of the form (y, H_i) in the sample for $i=0,1$. Let $f_{\text{adr}}(y) = H_0$ if $\hat{P}_{y,h,x}(H_0 \cap y) > \hat{P}_{y,h,x}(H_1 \cap y)$, and $f_{\text{adr}}(y) = H_1$ otherwise. Then we have $P[I(f_{\text{adr}}) > I(f^*)] < \delta$ given a sample of size $l = 1/\epsilon^2 \ln(2/\delta)$ where

$$\epsilon = \left[1 + \frac{|\hat{P}_{y,h,x}(H_0 \cap y) - \hat{P}_{y,h,x}(H_1 \cap y)|}{r} \right]^{1/2} - 1$$

for any $r > 2$. □

Note that computation of f_{adr} is very simple. This sample estimate, however, is not very useful since it uses the empirical measures that depend on the sample: δ is sensitive to both the size and the sample itself. Thus δ can be computed based on the sample, and the decision to employ this method can then be made based on performance measures of a competitive method. In Sec. 3, we apply this method to an example.

In the special case when $y^{(j)}$'s are independent, f_{adr} can be implemented by the test (Theorem 3.1 of Ref. 22)

$$\prod_{j=1}^{N} \hat{P}_{y,h,x}[H_0 \cap y^{(j)}] > \prod_{j=1}^{N} \hat{P}_{y,h,x}[H_1 \cap y^{(j)}]$$

such that the sample size is given by

$$l = \frac{1}{\epsilon^2} \ln(8n^2/\delta),$$

where

$$\epsilon = \left(1 + \frac{1}{r} \left\{ \prod_{j=1}^{N} \hat{P}_{y,h,x}[H_0 \cap y^{(j)}] - \prod_{j=1}^{N} \hat{P}_{y,h,x}[H_1 \cap y^{(j)}] \right\} \right)^{1/N} - 1.$$

Several variations of this technique have been studied in Refs. 12 and 16, which are both geared toward implementing a test using estimates for the required probabilities. Our method is geared toward minimizing the expected error and provides the sample size estimates as a function of a specified level of confidence.

2.3 *Nearest-Neighbor Rule*

We now consider a method based on the nearest-neighbor rule which is convenient to compute (unlike f_{emp}) and also yields a sample size that does not depend on the specific sample (unlike f_{adr}). This method also provides an alternative to popular methods based on sigmoidal feedforward networks while not incurring the high computational complexity of training the latter. As a side benefit, this method aids us in understanding the performance of the neural network approaches that are becoming increasingly popular.

We first show that the nearest-neighbor rule provides a good approximation to f^* when $\mathscr{F}$ is a set of rounded-off Lipschitz functions. We then consider a popular type of feedforward network that constitutes a subset of the required Lipschitz functions.

2.3.1 *Approximation of Lipschitz functions*

We assume that H_0 and H_1 are represented by 0 and 1 respectively, and we allow y to take continuous values from $[0,1]^N$. Given $(h_1, y_1), (h_2, y_2), \ldots, (h_l, y_l)$, for $y_i \in [0,1]^N$, $h_i \in \{0,1\}$, let

$$V(y_i) = \{ y \in [0,1]^N : \text{ for all } j, \|y - y_i\| \leq \|y - y_j\| \},$$

where $\|.\|$ is the Euclidean norm. We then have the Voronoi decomposition of $[0,1]^N$ into $V(y_i)$'s whose boundaries could have nonempty intersections. Let $\text{Int}[V(y_i)]$ denote the interior of $V(y_i)$, and let $h^{\#}_k(y_i)$, $k=0,1$, denote the number of h_l's in the sample that correspond to y_i with $h_l = k$. Now h_{NN} is defined by

$$h_{NN}(y) = \begin{cases} H_1 & \text{if } y \in \text{Int}[V(y_i)] \text{ and } h^{\#}_0(y_i) \leq h^{\#}_1(y_i) \\ H_0 & \text{if } y \in \text{Int}[V(y_i)] \text{ and } h^{\#}_0(y_i) > h^{\#}_1(y_i) \\ H_1 \text{ or } H_0 \text{ arbitrarily} & \text{if } y \in V(y_i) \cap V(y_j), \\ & i \neq j. \end{cases}$$

To facilitate our main result, we define two new quantities, the function $f_{NN}(.)$ and the cost functional $I_{NN}(.)$. For $f \in \mathscr{F}$, let f_{NN} be the nearest-neighbor rule based on the points $[h_1, f(y_1)], [h_2, f(y_2)], \ldots, [h_l, f(y_l)]$ as above [i.e., the same definition as that of $h_{NN}(.)$, except $f(y_i)$ replaces h_i]. Then we define the error functional

$$I_{NN}(f) = \int |h_{NN}(y) - f_{NN}(y)| dP_{y|x} dP_{h|x} dP_x,$$

where $y = (y^{(1)}, y^{(2)}, \ldots, y^{(N)})$. Let f^*_{NN} minimize $I_{NN}(f)$ over $\mathscr{F}$. We consider $\mathscr{F}_t$ which are "rounded-off" versions as defined below so that Eq. (1) can still be used for the expected error.

Theorem 2.3 Let $\mathscr{F} = \{f : [0,1]^N \mapsto [0,1]\}$ denote the class of Lipschitz functions with the constant k, i.e. for all $f \in \mathscr{F}$, we have $|f(x) - f(y)| \leq k\|x - y\|$ for all $x, y \in [0,1]^N$. Let $\mathscr{F}_t = \{f_t : f \in \mathscr{F}\}$, $0 < t < 1/2$, where

$$f_i(y)=\begin{cases}0 & \text{if } f(y)\leq 1/2-t\\ 1 & \text{if } f(y)\geq 1/2+t\\ 0 \text{ or } 1 \text{ arbitrarily} & \text{otherwise}\end{cases}$$

Given a sample of size at least

$$\frac{k^2N^3}{\delta t^2\epsilon^2}+2$$

we have $P[I(h_{NN})-I(f^*)>2\epsilon]<\delta$, where $I(f^*)=\min_{f\in\mathscr{F}_l}I(f)$.

Proof: We first show that $P[\sup_{f\in\mathscr{F}_l}|I(f)-I_{NN}(f)|>\epsilon]<\delta$ for the sample size given in the statement of the theorem, which implies $P[I(f^*_{NN})-I(f^*)>2\epsilon]<\delta$ by the argument of Ref. 11. This in turn implies the theorem since $I_{NN}(h_{NN})=0=\min_{f\in\mathscr{F}}I_{NN}(f)$ and $h_{NN}\in\mathscr{F}_l$. Let us define

$$C_\alpha=\{(y_1,y_2,\ldots y_l)\in[0,1]^{lN}:\ \max_{1\leq i\leq l}P[V(y_i)]>\alpha\}.$$

We have

$$P[C_\alpha]\leq P\{(y_1,y_2,\ldots y_l)\in[0,1]^{lN}:\text{there is } i \text{ such that for all } j\neq i, y_j\notin\tilde{D}^\alpha(y_i)\},$$

where $\tilde{D}^\alpha(y_i)$ is a ball with the smallest radius that contains $V(y_i)$, measuring at least α. $D^\alpha(y_i)$ is a ball centered at y_i with smallest the radius measuring at least α. Now we have

$$\begin{aligned}P(C_\alpha)&\leq lP\{(y_1,y_2,\ldots y_l)\in[0,1]^{lN}:\text{there is } i \text{ such that for all } j\neq i, y_j\notin D^\alpha(y_i)\}\\&\leq lP\{(y_1,y_2,\ldots y_l)\in[0,1]^{lN}:\text{there is } \xi\in[0,1]^N \text{ such that for all } j, y_j\notin D^\alpha(\xi)\}\\&\leq l(1-\alpha)^{l-1}.\end{aligned}$$

By choosing $\alpha=(1/l-1)\ln(l/\delta)$, and using the fact that $(1-\beta/n)^n<e^{-\beta}$ this probability is no more than δ.

Let us project the y_i's onto to each axis $j=1,2,\ldots,N$ and identify for each y_i the interval I^j that contains the projection of y_i and contains all the points of the axis that are closest to the projection of y_i than a projection of any other point. We now classify y_i's into two classes based on the condition $\max_j|I^j_i|<(t/\sqrt{Nk})$, where $|I^j_i|$ is the length of the interval I^j_i. If the condition is satisfied for y_i, the radius of $\tilde{D}(y_i)$ is less than or equal to t/k, where $\tilde{D}(y_i)$ is a ball with the smallest radius that contains $V(y_i)$; consequently for this y_i, we have $|f(x)-f_{NN}|(x)\leq t$. If the condition is not satisfied for y_k, then for some j we have $|I^j_k|>(t/\sqrt{Nk})$, which implies that the radius of $\tilde{D}(y_i)$ is at least $(t/\sqrt{Nk})$. Now there are no more than $(kN^{3/2}/t)$ of i's with the above condition not satisfied [because for each axis j, there are no more than $(k\sqrt{N}/t)$ such y_i's]. Thus we have with a probability at least $1-\delta$, we have,

$$P[\{y:|f(y)-f_{NN}(y)|\geq t\}]\leq\frac{k\alpha}{t}=\frac{kN^{3/2}}{t(l-1)}\ln(1/\delta).$$

The right-hand side can be made smaller than ϵ under the condition

$$t(l-1)\epsilon\geq kN^{3/2}\ln(l/\delta),$$

which is satisfied under the condition

$$t(l-1)\epsilon\geq kN^{3/2}\sqrt{l/\delta}$$

by using $\ln(x)\leq\sqrt{x}-1$ for $x>1$. This condition is satisfied under the condition

$$l\geq\frac{k^2N^3}{t^2\epsilon^2\delta}+2,$$

which is satisfied under the hypothesis since the condition $l>b^2+2$ implies $l\geq b\sqrt{l}+2$. □

This theorem indicates that the error of the nearest-neighbor rule is within 2ϵ of $I(f^*)$ with a probability of $1-\delta$ when $\mathscr{F}_l$ corresponds to rounded-off Lipschitz functions. This method is implemented in Sec. 3 for an example.

2.3.2 *Feedforward neural networks*

We now show that $h_{NN}(.)$ of the last section is a good approximation to the best possible neural network and also ensures ease of computation. We consider the popular class of feedforward sigmoidal networks which seem to be very widely used.

A general architecture of a multilayer feedforward network consists of an input layer with d units and an output layer with m units, and one or more hidden layers. Consider a network with a single hidden layer and single output node ($m=1$). The hidden unit j has a weight vector $b_j\in\Re^d$ and a threshold $t_j\in\Re$. The output of the j'th hidden unit is $\sigma(b_j^Ty-t_j)$, where $y=(y^1,y^2,\ldots,y^d)$ is the input vector, b_j^Ty denotes the scalar product, and $\sigma:[0,1]\mapsto[0,1]$ is called an activation function. The output of the neural network is given by

$$f(y)=\sum_{j=1}^{M}a_j\sigma(b_j^Ty+t_j),$$

where $a=(a_1,a_2,\ldots,a_M)$ is the weight vector of the output node, M is the number of neurons in the hidden layer, and w is the parameter or weight vector of the network that consists of a, $b_1,b_2,\ldots,b_M$ and $t_1,t_2,\ldots,t_M$. We consider sigmoidal hidden units of the form $\sigma(x)=1/(1+e^{-\rho z})$, for $\rho,z\in\Re$. The Lipschitz constant of these networks is given by

$$k=\frac{\gamma}{4}ab\sqrt{M},$$

where $a=\max a_i$ and $b=\max b_{ij}$. Let $\mathcal{N}$ be the set of all neural networks of this type with fixed M. Let $\mathcal{N}_t$ correspond to the rounded-off version of $\mathcal{N}$ as in Theorem 2.3 and let f_N^* minimize Eq. (1) over $\mathcal{N}_t$. Then from Theorem 2.3, for the sample size $\{\gamma^2a^2b^2Md^3/16\,\delta t^2\epsilon^2\}+2$ we have $P[I(f_{NN})-I(f_N^*)>2\epsilon]<\delta$. Thus f_{NN} provides a good approximation to f_N^*. We note that f_N^* is hard to compute even when the distributions are known, but f_{NN} is easy to compute based on a sample and requires no knowledge of the distributions.

Since f_N^* is not computable in our case, an empirical approximation to it can be examined. One can consider f_{emp} for $\mathcal{F}=\mathcal{N}_t$, which minimizes the empirical error as described in Sec. 2.1. But since $\mathcal{N}_t$ is not finite, Theorem 2.1 is not directly applicable. A generalization to Theorem 2.1 is feasible so the f_{emp} approximates f_N^*, but the computation of f_{emp} is intractable[23,24] whereas the nearest-neighbor rule offers the same type of approximation to f^* while being computable in linear time in the sample size.

3 Example

We consider a system with five detectors, $D_1,\ldots,D_5$. The hypotheses H_1 and H_0 are generated with equal probabilities for any x. This behavior is simulated by generating† a uniformly distributed random variable over the interval [0,1] and checking to see if it lies in the interval [0, 0.5]. Each detector is given the hypothesis as input and it produces an output that disagrees with the input according to a probabilistic strategy. The detector D_i, $i=1,2,\ldots,5$ introduces an error as follows: with a probability of $1-i/10$ it passes on the input to output, and with a probability $i/10$ it passes on the opposite of the input. The individual detector behavior is implemented by generating a uniform random variable in the range $[0,D]$ and checking if it falls in the interval $[0,iD/10]$. It is assumed that the pseudo random number generator yields independent outputs and the error processes of the individual detectors are probabilistically independent.

A sequence of examples has been generated and given as input to the Bayesian fuser, approximate decision rule, and the nearest-neighbor rule. The Bayesian fuser is implemented by using the analytical formulas for the distribution of errors; here independence between the various detectors is assumed to obtain the Bayesian rule. For the approximate decision rule, the probabilities are estimated based on the sample seen so far. The nearest-neighbor rule is based on the sample seen so far. Each example is given as input to the three fusers and their outputs are computed. The percentage of correctly classified examples (among the sample seen so far) is computed and shown in Fig. 2 and Table 1. The plots in Figs. 2(a), 2(b), and 2(c) are for 1,000, 10,000 and 100,000 examples respectively. The performances of the approximate decision rule and nearest-neighbor rule showed overall improving trends as the training progressed;

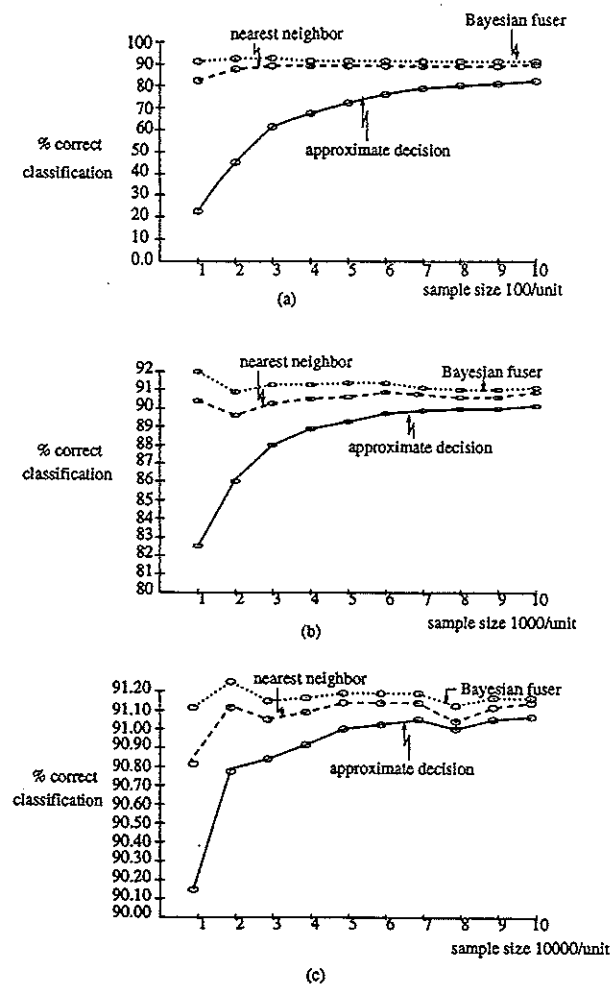


Fig. 2 Relative performance of Bayesian fuser, approximate decision rule, and nearest-neighbor rule.

the performance of the latter was consistently better during the entire training process. After 1,000 examples, the approximate decision rule and nearest-neighbor rule achieved performances within 10 and 2% respectively of that of the Bayesian fuser, as shown in Fig. 2(a). After 10,000 examples, the performances of the approximate decision rule and nearest-neighbor rule were within 1 and 0.3% respectively of that of the Bayesian fuser [Fig. 2(b)]. After 100,000 examples, the performances of the approximate decision rule and nearest-neighbor rule were within 0.096 and 0.03% respectively of that of Bayesian fuser [Fig. 2(c)]. This simulation has been repeated with different starting seeds for the pseudo random number generator with almost identical qualitative behavior. The training program has been implemented on an SPARC workstation IPX; the total execution time for the entire simulation is 2 days when executed as a background process. The superior performance of the nearest-neighbor rule over the approximate decision rule appears to be an artifact of the present example—no general results are known that indicate the unilateral superiority of the former.

†All simulations are carried out using pseudo random number generators.

Table 1 Percentage of correct classification by Bayesian fuser, approximate decision rule, and nearest-neighbor rule.

number of samples	Bayesian fuser	approximate decision rule	nearest neighbor rule
100	91.9192	23.0000	82.8283
200	93.0000	47.0000	87.0000
300	92.3333	61.6667	88.3333
400	91.5000	68.5000	88.5000
500	91.2000	72.8000	88.8000
600	91.3333	76.0000	89.0000
700	91.1429	78.1429	89.2857
800	91.1250	79.7500	89.3750
900	91.5556	81.2222	89.8889
1000	91.9920	82.5826	90.3904
2000	90.8000	86.0000	89.6000
3000	91.2333	88.0333	90.2333
4000	91.2000	88.8000	90.4500
5000	91.2800	89.3600	90.6800
6000	91.3167	89.7167	90.8167
7000	91.1857	89.8143	90.7571
8000	91.0125	89.8125	90.6375
9000	90.9889	89.9222	90.6556
10000	91.1100	90.1500	90.8100
20000	91.2650	90.7850	91.1150
30000	91.1500	90.8300	91.0500
40000	91.1675	90.9275	91.0925
50000	91.1900	90.9980	91.1300
60000	91.1867	91.0267	91.1367
70000	91.1800	91.0429	91.1371
80000	91.1263	91.0062	91.0887
90000	91.1622	91.0556	91.1289
100000	91.1650	91.0690	91.1350

In some cases, typically when independence is not satisfied, the problem of implementing a Bayesian fuser could be computationally expensive (since the problem could be NP-complete). In such cases, information about the *a priori* probabilities and error distributions can be used to generate examples by using Monte Carlo methods. The approximate decision rule and the nearest-neighbor rule can be computed based on the examples generated (as shown in the above example). The utility of such methods depends on the ease with which the examples can be generated and the characteristics of the programs that generate the pseudo random variables.

4 Conclusions

We have studied the problem of decision fusion in distributed detection systems when training examples are available but no information is available about the probability of errors committed by the individual detectors. Since no information about the underlying probabilities is available, an exact implementation of the optimal rule is not possible based on a finite set of examples. We describe three methods that, given a sufficiently large training sample, yield a fusion rule that approximates an optimal one with an arbitrarily high level of confidence. The three methods are (i) empirical estimation, (ii) approximate decision rule, and (iii) nearest-neighbor rule. We show that a nearest-neighbor rule provides a computationally viable solution. The proposed method is useful in systems where either the underlying probabilities are not known or the optimal rule is too difficult to implement due to computational complexity.

There are two generalizations of the formulation studied here. Rao[16] discusses the estimation of fusion rules in multiple sensor systems, and Rao and Oblow[21] discuss empirical implementation of optimal fusion rules for a system of probably approximately correct learners. These results are more general in that $\mathscr{F}$ is not finite and consequently the sample size estimates are presented in terms of the Vapnik and Chervonenkis dimension.[11]

Several directions can be pursued for future work. It would be interesting to investigate computational methods for fusion rule estimation based on wavelets and traditional regression estimates. The sample sizes presented in this paper are only sufficient, but lower bounds on the sample sizes provide a better understanding of the complexity of the underlying problem.

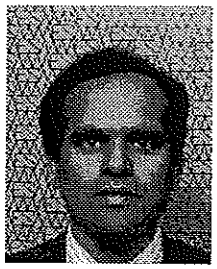
Acknowledgments

The research by Nageswara S. V. Rao was sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract no. DE-AC05-84OR21400 with Lockheed Martin Energy Systems, Inc. The research by S. Sitharama-Iyengar was funded by the U.S. Office of Naval Research under Grant no. N0014-91J-1306 and N0014-85K-0611. The authors are grateful to the two anonymous reviewers for their constructive comments, which improved the quality of this paper.

References

1. B. Grofman and G. Owen, Eds., *Information Pooling and Group Decision Making*, JAI Press, Greenwich, CT (1986).
2. J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds., pp. 43–98, Princeton Univ. Press, Princeton, NJ (1956).
3. C. W. J. Granger, "Combining forecasts—twenty years later," *J. Forecasting* **8**, 67–173 (1989).
4. C. K. Chow, "Statistical independence and threshold functions," *IEEE Trans. Electronic Computers* **EC-16**, 66–68 (1965).
5. S. Hashem, B. Schmeiser, and Y. Yih, "Optimal linear combinations of neural networks: An overview," in *Proc. 1994 IEEE Conf. on Neural Networks*, pp. 1507–1512 (1994).
6. B. V. Dasarathy, *Decision Fusion*, IEEE Computer Society Press, Los Alamitos, CA (1994).
7. Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerospace Electronic Systems* **22**(1), 98–101 (1986).
8. S. S. Iyengar, H. Min, and L. Prasad, *Advances in Distributed Sensor Integration: Application and Theory*, Prentice-Hall, Englewood Cliffs, NJ (1995).
9. S. C. A. Thomopoulos, R. Viswanathan, and B. K. Bougoulias, "Optimal and suboptimal distributed decision fusion," *IEEE Trans. Aero*
10. E. Drakopoulos and C. C. Lee, "Optimal multisensor fusion of correlated local decision," *IEEE Trans. Aerospace Electronics Systems* **27**(4), 593–605 (1991).
11. V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, New York (1982).
12. A. Naim and M. Kam, "On-line estimation of probabilities for Bayesian distributed detection," *Automatica* **30**(4), 633–642 (1994).

13. N. S. V. Rao, S. S. Iyengar, and R. L. Kashyap. "Computational complexity of distributed detection problems with information constraints," *Comp. Elec. Eng.* **19**(6), 445–451 (1993).
14. J. N. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," *IEEE Trans. Automatic Control* **AC-30**(5), 440–446 (1985).
15. B. L. S. Prakasa Rao, *Nonparametric Functional Estimation*, Academic Press, New York (1983).
16. N. S. V. Rao, "Fusion rule estimation in multiple sensor systems with unknown noise densities," *J. Franklin Inst.* **331B** (5), 509–530 (1995).
17. L. G. Valiant, "A theory of the learnable," *Commun. ACM* **27**(11), 1134–1142 (1984).
18. B. K. Natarajan, *Machine Learning: A Theoretical Approach*, Morgan Kaufmann, San Mateo, CA (1991).
19. P. Billingsley, *Probability and Measure*, 2nd ed., Wiley, New York (1986).
20. W. A. Hashlamoun and P. K. Varshney, "Further results on distributed Bayesian signal detection," *IEEE Trans. Information Theory* **39**(5), 1660–1661 (1993).
21. N. S. V. Rao and E. M. Oblow, "*N*-learners problem: System of PAC learners," in *Computational Learning and Machine Learning*, MIT Press, Cambridge, MA (in press).
22. N. S. V. Rao, "Distributed decision fusion using empirical estimation," *IEEE Trans. Aerospace Electronic Systems* (in press).
23. A. L. Blum and R. L. Rivest, "Training a 3-node neural network is NP-complete," *Neural Networks* **5**, 117–127 (1992).
24. V. Roychowdhury, K. Siu, and A. Orlitsky, Eds., *Theoretical Advances in Neural Computation and Learning*, Kluwer Academic Pub., Amsterdam (1994).



Nageswara S. V. Rao received a BTech in electronics and communications engineering from the Regional Engineering College, Warangal, India, in 1982; an ME from the School of Automation, Indian Institute of Science, Bangalore, India, in 1984; and a PhD in computer science from Louisiana State University, Baton Rouge, in 1988. He has been a research staff member at the Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, Tennessee, since 1993. He is an adjunct associate professor in the Department of Computer Science, Old Dominion University, Norfolk, Virginia, where he was an assistant professor from 1988 until 1993. His research interests include robot navigation, multiple sensor systems, *N*-learner systems, fault diagnosis and adaptive algorithms. He has published more than 90 research articles in various journals, conference proceedings and books.

S. Sitharama Iyengar: Biography and photograph appear with the paper "New computational technique for complementary sensor integration in detection-localization systems" in this issue.